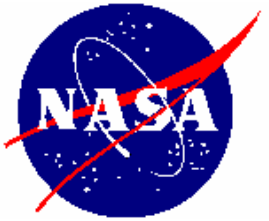


And some have said “software isn’t critical”

Leadership ViTS Meeting
13 February 2006

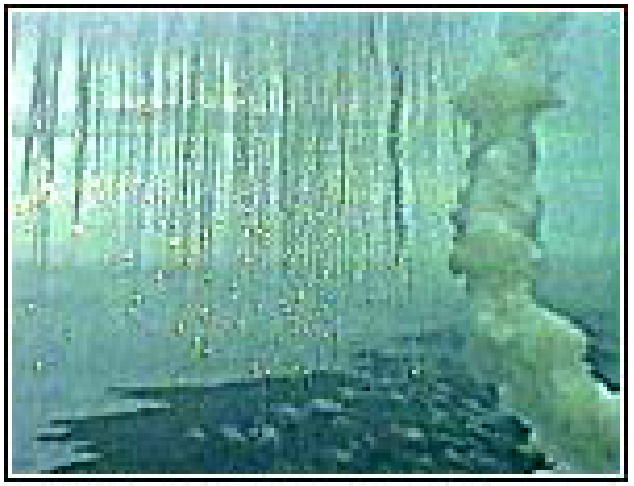
Jim Lloyd, Deputy Chief
Office of Safety and Mission Assurance

The Failure of the Ariane 5's First Launch

On June 4, 1996 the European Space Agency (esa) launched an Ariane 5 rocket from Kourou, French Guiana. The rocket was destroyed forty seconds after its lift-off.

According to the report written by the Inquiry Board (published 19 July 1996) the proximate cause of the loss of the Ariane 501 was the complete loss of guidance and attitude information 37 seconds after main engine ignition sequence start (or about 30 seconds after lift-off).

http://www.ima.umn.edu/~arnold/disasters/ariane.html



Smoke from the explosion
June 4, 1996 (AP Photo)

- The launch of the Ariane 5 was its first, after a decade of development costing over $7 billion.
- The destroyed rocket and its cargo were valued at $500 million.

Key Differences in Ariane Inertial Reference Systems

picture courtesy of esa

- The inertial reference system (IRS) of Ariane 5 is essentially common to its predecessor, Ariane 4 and, for this reason, its software (SW) was reused.
- The SW is used before launch to align the inertial reference system.
- It operates for 40 seconds after launch
- In Ariane 4, the SW also enables a rapid realignment of the system in case of a hold late in the countdown.
- On Ariane 5, this rapid realignment function does not serve any purpose, nevertheless, it had been retained for commonality reasons.
- As it was later discovered in the inquiry, this caused the interruption in the inertial reference system computers.

Accident Sequence

- A critical piece of software had been reused from the Ariane-4 system, but behaved differently in the Ariane-5 because of differences in the operational parameters of the two rockets.
- During a data conversion from a 64-bit value to a 16-bit value for the horizontal component of the velocity vector with respect to the platform, an overflow occurred, resulting in a conversion failure and submitting a failure diagnostic code as an input to the IRS.
- As the code was not designed to handle such an error, the first leg of the redundant inertial reference system simply shut down.
- As programmed, the control passed to a second inertial reference system, which, operating under the same information as the first, also (predictably) shut down!
- The failure of these two systems led to the on-board computer misinterpreting diagnostic data as proper flight data, signaling a need for correction to a deviation in flight path that was not really happening.
- A rather abrupt, but commanded, correction to the perceived deviation in the flight path created aerodynamic forces sufficient to rip the boosters from the rocket while simultaneously activating the rocket's self-destruct mechanism.

What Went Wrong with Software Reuse?

Re-evaluation and analyses of parameters from one system to the next (Ariane 4 to 5) not understood– need to evaluate thoroughly deemed "not necessary"

- Have specific software qualification review, with independent assurance, for each item of equipment incorporating software.
- Identify safety critical software and manage its configuration.
- Perform analyses on changes to software, especially the safety critical software.
  - Changes to safety critical software or systems that impact critical functions need to be reviewed by a group of external experts and reported to a Qualification Board/Safety Panel.
- Improve overall coordination of analysis relating to software.
- Remove, switch off, or inhibit un-used software.

Limited Review Process: the validation of design decisions and flight qualification was limited by agreement of all major partners in the Ariane 5 program, resulting in the alignment software not being fully analyzed.

- Review all flight software.
- Check documented specification and code value assumptions against the actual parametric ranges of the equipment.
- Verify the range of values taken by any variables in the software.
- Improve system qualification environment through systematic use of real equipment and components (rather than simulators) wherever possible.

What Went Wrong with Software Reuse?

Insufficient Testing at the System Level: The tests were mainly performed at the equipment level without comprehensive testing or even simulation at the sub-system and system levels.

- Seek broader test coverage.
- Perform complete, closed-loop, system testing with realistic input data.
- Introduce overlaps and deliberate redundancy between successive tests:
  - at equipment level,
  - at stage level,
  - at system level.

Incomplete/Insufficient Simulations Performed: While feasible, it was decided to use only the simulated output of the inertial reference system, not the system itself or its detailed and validated simulation.

- Simulate using real data and equipment when available.
- Complete end-to-end simulation testing must take place before the mission.

Resolving any of these 4 findings could have detected the potential for failure.

Software and Associated Data Handling need to be Evaluated Carefully

Problems associated with software and associated data handling, with its potential for wreaking havoc if not treated with respect, continue as attested by these recent news items indicate:

1/3/2006 Ten JAL Flights Delayed up to One Hour at Tokyo Airport Due to Computer Glitch

1/3/2006 Credit Card Glitch Double-bills Customers

1/6/2006 Computer Glitch Affects United Airlines

1/9/2006 Patients Put at Risk by NHS Computer Fault

1/16/2006 Social Security Database Compromised

1/30/2006 Newly Launched Japanese Satellite Back to Normal After Computer Glitch

Oversights in software development affect each of us daily.